#12

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant:	Douglas L. Rollins	§	Group Art Unit:	2122
		§		
Serial No.:	09/272,845	§		
		§	Examiner:	Chuck O. Kendall
Filed:	March 19, 1999	§		
		§		
For:	Software Module Update	§	Atty. Dkt. No.:	MCT.0088US (MUEI-0471.00/US)
		§		

Board of Patent Appeals & Interferences
Commissioner for Patents
Washington, D.C. 20231

REPLY BRIEF

Dear Sir:

The following reply is submitted to the Examiner's Answer.

I. CLAIMS APPEALED

The appealed claims are claims 1-16, 41 and 42, grouped together; and claims 17-32, 43 and 44, grouped together.

II. REPLY TO EXAMINER'S ARGUMENTS

To establish a *prima facie* case of obviousness in view of certain references, there must be a suggestion or motivation in the prior art for the combination of these references. However, the Examiner still fails to show specific support for the alleged suggestion or motivation for the combination of Apfel and Furner and thus, still fails to establish a *prima facie* case of

Date of Deposit: February 3, 2003
I hereby certify under 37 CFR 1.8(a) that this correspondence is being deposited with the United States Postal Service as **first class mail** with sufficient postage on the date indicated above and is addressed to the Board of Patent Appeals & Interferences, Commissioner for Patents, Washington, DC 20231.
Debra Cutrona
Debra Cutrona

obviousness. *See, for example, Ex parte Gambogi*, 62 USPQ2d 1209, 1212 (Bd. Pat. App. & Int. 2001); *In re Rijckaert*, 28 USPQ2d 1955, 1957 (Fed. Cir. 1993); M.P.E.P. § 2143.

On page 5 of the Examiner's Answer, the Examiner states, "both Apfel and Furner disclose providing more current versions as needed or when available, and thus are analogous art," and further states, "therefore, one would be motivated to make that combination to implement such as Appellant's claimed invention." Examiner's Answer, 5. However, the mere fact that two references may be classified as belonging to analogous art is not sufficient to establish a *prima facie* case of obviousness, as the prior art must supply a suggestion or motivation to combine the references. It is noted that the Examiner's allegation that Furner discloses providing more current versions of software as needed is inaccurate and mischaracterizes Furner, as further discussed below.

The Examiner contends that Furner is only relied on for purposes of associating circuitry with software. Examiner's Answer, 3. If this is the case, then a *prima facie* case of obviousness has not been established for the additional, independent reason that Apfel fails to disclose determining whether a second version of a software module is compatible with circuitry that is associated with the software module. The Examiner contends that Apfel teaches this limitation and cites language from lines 15-25 in column 7 of Apfel. However, the Examiner's quotation is misleading, as the Examiner selectively inserts language into the quoted language. The exact quotation is as follows:

> For example, even if an upgrade is available, it should not be downloaded if the computer 20 already has

> the upgrade or if the upgrade is somehow incompatible with computer 20. Providing the URL may include load balancing between servers because if a primary server has a high volume, URLs of alternate servers with the same upgrade package may be utilized.

Apfel, 7:15-25. Thus, in the cited passage, Apfel does not discuss circuitry as being incompatible with a software upgrade, but rather, this incompatibility is being read into Apfel by the Examiner. It is noted that one skilled in the art may recognize that the incompatibility Apfel is referring to the incompatibility of upgrade software to other software of the computer, such as application software and/or operating system software, for example. The language does not state that the incompatibility is with hardware or circuitry of the computer system. A reading of Apfel in its entirety reveals that Apfel is concerned with ensuring compatibility with software and is not concerned with compatibility of software with circuitry. Thus, Apfel does not teach or suggest, as contended by the Examiner, determining whether a second version of a software module is compatible with circuitry that is associated with the software module. Therefore, if the Examiner is relying on Apfel for this teaching, a *prima facie* case of obviousness has not been established for this additional, independent reason.

In the Appeal Brief, Applicant stated that Furner did not teach or suggest determining whether a second version of a driver is more current than a first version of a driver. In the Examiner's Answer, the Examiner takes the contrary position, stating that Furner is "concerned with whether a second version of a driver is more current than a first version of a driver (software)." Examiner's Answer, p. 3. In support of this position, the Examiner cites language from lines 45-55 in column 1 of Furner and cites languages from lines 62-67 in column 4 of Furner. However, the Examiner is mischaracterizing Furner. More specifically, referring to the cited passages, in lines 45-55 in column 1 of Furner, Furner states:

> These hardware devices are connected to the computer system through one or more input/output (I/O) buses, which may be of many different types including, for example, the well-defined ISA, EISA, and PnPISA buses.
>
> For each additional hardware device installed in the computer system, an associated driver is loaded into the computer memory, and is used by the operating system and application programs to communicate with the associated hardware device.

Furner, 1:45-55. In this cited language, Furner discusses different versions of hardware devices to accommodate different revisions of a device model. However, this language does not teach or even suggest determining whether a second version of a driver is more current than a first version of a driver, as the cited language is only concerned with *hardware* revisions. Thus, the Examiner must rely on Apfel to supply this missing limitation. In support of Furner allegedly teaching determining whether a second version of a driver is more current than a first version of a driver, the Examiner also cites language from line 62-67 in column 4 of Furner, reproduced below:

> The system comprises means for determining a relative capability of each the plurality of drivers to support the hardware instance, and means for selecting an optimal driver having a capability to support the hardware instance that is greater than a capability of other drivers to support the hardware instance.

Furner, 4:62-67. As can be seen from this cited language, Furner is discussing the compatibility of hardware to a driver. However, Furner neither teaches nor suggests in this language determining whether a second version of a driver is more current than a first version of a driver.

The Examiner also contends, "it is well known in the art for hardware to be operative utilizing associated device drivers (software) for its efficient operation, and revising hardware would also be directly associated with revising the device driver." Examiner's Answer, p. 4. It appears the Examiner is relying on the general level of skill in the art (in view of Furner) to derive the limitation of determining whether a second version of driver is more current than a first version of a driver. This is improper, as "rarely, however, will the skill in the art component operate to supply missing knowledge or prior art to reach an obviousness judgment." *Al-Site Corp. v. VSI Int'l, Inc.*, 50 USPQ2d 1161, 1171 (Fed. Cir. 1999). In short, the Examiner points to no language in Furner that teaches or suggests determining whether a second version of driver is more current than a first version of a driver.

In conclusion, Furner neither teaches nor suggests determining whether a second version of a driver is more current than a first version of a driver; and Apfel neither teaches nor suggests determining whether a second version of a software module is compatible with circuitry that is associated with this software module. Thus, the Examiner must rely on the piecemeal combination of Apfel and Furner to derive the claimed invention. However, the Examiner fails to show any support for the alleged suggestion or motivation to combine these two references to derive the claimed invention. Therefore, for at least these reasons, a *prima facie* case of obviousness has not been established for claims 1-32 and 41-44.

Thus, the rejections of the claims should be reversed.

The Commissioner is authorized to charge any additional costs associated with this Reply Brief to Deposit Account No. 20-1504 (MCT.0088US)

Respectfully submitted,

Date: February 3, 2003

Fred G. Pruner, Jr., Reg. No. 40,779
TROP, PRUNER & HU, P.C.
8554 Katy Freeway, Suite 100
Houston, TX 77024-1805
713/468-8880 [Phone]
713/468-8883 [Facsimile]

21906
PATENT TRADEMARK OFFICE

APPENDIX OF CLAIMS

The claims on appeal are:

1. A method comprising:

identifying a first version of a software module, the software module being installed on a computer system and being associated with circuitry of the computer system;

identifying a second version of the software module;

automatically determining whether the second version is more current than the first version and whether the second version is compatible with the circuitry; and

indicating the result of the determination.

2. The method of claim 1, further comprising obtaining that version of the software module determined to be the most current version.

3. The method of claim 2, further comprising loading the obtained version of the software module.

4. The method of claim 1, wherein the act of identifying a first version of the software module comprises communicating with a physical device associated with the software module.

5. The method of claim 4, wherein the act of communicating comprises determining an identifier value of the physical device.

6. The method of claim 5, further comprising determining a subsystem identifier value of the physical device.

7. The method of claim 4, wherein the act of communicating comprises determining a basic input-output system version identifier value of the physical device.

8. The method of claim 1, wherein the act of identifying a second version of the software module comprises communicating with an update information source.

9. The method of claim 8, wherein the act of communicating further comprises:

identifying the software module to the update information source;

receiving, from the update information source, an indication of the second version of the software module.

10. The method of claim 8, wherein the act of communicating comprises communicating by a modem.

11. The method of claim 8, wherein the act of communicating comprises communicating by a computer network.

12. The method of claim 8, wherein the act of communicating comprises communicating with a database server device.

13. The method of claim 1, wherein the act of determining comprises comparing at least one characteristic of the first identified first version of the software module with the same characteristic of the second identified first version of the software module.

14. The method of claim 1, wherein the act of indicating comprises visually displaying an indication of the software module determined to be the most current version to a user.

15. The method of claim 2, wherein the act of obtaining comprises retrieving that version of the software module determined to be the most current version from an update source.

16. The method of claim 15, wherein the act of retrieving comprises retrieving from an update source that is physically distinct from the location of the first identified version of the software module.

17. A program storage device, readable by a programmable control device, comprising:

instructions stored on the program storage device for causing the programmable control device to

identify a first version of a software module, the software module being installed on a computer system and being associated with the circuitry of the computer system;

identify a second version of the software module;

automatically determine whether the second version is more current than the first version and whether the second version is compatible with the circuitry; and

indicate the result of the determination.

18. The program storage device of claim 17, further comprising instructions to obtain that version of the software module determined to be most current version.

19. The program storage device of claim 18, further comprising instructions to load the obtained version of the software module.

20. The program storage device of claim 17, wherein the instructions to identify a first version of the software module comprise instructions to communicate with a physical device associated with the software module.

21. The program storage device of claim 20, wherein the instructions to communicate comprise instructions to determine an identifier value of the physical device.

22. The program storage device of claim 21, further comprising instructions to determine a subsystem identifier value of the physical device.

23. The program storage device of claim 20, wherein the instructions to communicate comprise instructions to determine a basic input-output system version identifier value of the physical device.

24. The program storage device of claim 17, wherein the instructions to identify a second version of the software module comprise instructions to communicate with an update information source.

25. The program storage device of claim 24, wherein the instructions to communicate further comprise instructions to:

identify the software module to the update information source;

receive, from the update information source, an indication of the second version of the software module.

26. The program storage device of claim 24, wherein the instructions to communicate comprise instructions to communicate by a modem.

27. The program storage device of claim 24, wherein the instructions to communicate comprise instructions to communicate by a computer network.

28. The program storage device of claim 24, wherein the instructions to communicate comprise instructions to communicate with a database server device.

29. The program storage device of claim 17, wherein the instructions to determine comprise instructions to compare at least one characteristic of the first identified first version of the software module with the same characteristic of the second identified first version of the software module.

30. The program storage device of claim 17, wherein the instructions to indicate comprise instructions to visually display an indication of the software module determined to be the most current version to a user.

31. The program storage device of claim 18, wherein the instructions to obtain comprise instructions to retrieve that version of the software module determined to be the most current version from an update source.

32. The program storage device of claim 31, wherein the instructions to retrieve comprise instructions to retrieve from an update source that is physically distinct from the location of the first identified version of the software module.

41. The method of claim 1, wherein the determining comprises:

determining a date associated with the first version, the date establishing compatibility between the second and first versions.

42. The method of claim 1, wherein the circuitry comprises an add-in card.

43. The program storage device of claim 17, further comprising instructions to cause the programmable control device to determine a date associated with the first version, the date establishing compatibility between the first and second versions.

44. The programmable storage device of claim 17, wherein the circuitry comprises an add-in card.